

Listing of Claims

The following claims supersede the previous claims.

Claim 1. (canceled)

Claim 2. (currently amended) A buffer cache having an efficient utilization characteristic, comprising: as recited in claim 1, wherein:

a first data buffer, the first data buffer being is configured with a state of completion of processing;

a first descriptor buffer configured with a state of completion of processing,  
the state of completion of processing of the first descriptor buffer being is  
independent of the state of completion of processing of the first data buffer;  
~~and~~

a free data buffer list configured to identify only the first data buffer in a free condition, the free condition of the first data buffer being independent of the state of completion of processing of the first descriptor buffer; and

the buffer cache being is configured to place the first data buffer on the free data buffer list according to the state of completion of processing of only the first data buffer.

Claim 3. (original) A buffer cache as recited in claim 2, wherein the buffer cache is configured so that the placing of the first data buffer on the free data buffer list may occur immediately upon the state of completion of processing of the first data buffer becoming a processing-complete state.

Claim 4. (currently amended) A buffer cache having an efficient utilization characteristic, as recited in claim 1, further comprising:

a first data buffer;

a first descriptor buffer configured with a state of completion of processing;

a free data buffer list configured to identify only the first data buffer in a free condition, the free condition of the first data buffer being independent of the state of completion of processing of the first descriptor buffer;

a free descriptor buffer list; and wherein:

the buffer cache is configured to place the first descriptor buffer on the free descriptor buffer list according to the state of completion of processing of only the first descriptor buffer; and

the buffer cache is further configured to respond to a data access request by taking the first data buffer from the free data buffer list and taking the first descriptor buffer from the free descriptor buffer list, each taking being separate from the other taking.

Claim 5. (currently amended) A buffer cache as recited in claim 4, wherein:

the first descriptor buffer is configured with a descriptor having a first value that identifies the first data buffer as a particular data buffer that is associated with the first descriptor buffer; and

the first descriptor buffer is configured with the descriptor having a second value that identifies the first data buffer as a particular data buffer that is on the free data buffer list, the second value being independent of the first value.

Claim 6. (currently amended) A buffer cache as recited in claim 5, the buffer cache further comprising wherein :

a second data buffer; wherein:

the buffer cache is further configured to change the first value to terminate that association, the change being independent of the state of completion of processing of the first descriptor buffer;

the buffer cache is configured with a second descriptor buffer having a descriptor having a third value that identifies the second data buffer as a particular data buffer that is associated with the second descriptor buffer; ~~and~~

the buffer cache is configured with a the second ~~free~~ descriptor buffer having a the descriptor having a fourth value that identifies the second data buffer as a particular data buffer that is on the free data buffer list, the fourth value being independent of the third value;

the buffer cache is further configured to change the third value of the descriptor of the second descriptor buffer to make an association between the first data buffer and the second descriptor buffer; and

the buffer cache is further configured to respond to the second and third ~~fourth~~ values and take the first data buffer and the second descriptor buffer in response to a data access request for data not already stored in the buffer cache, that taking being independent of the state of completion of processing of the first descriptor buffer.

Claim 7. (currently amended) A buffer cache as recited in claim 2 ~~1~~, wherein ~~the first descriptor buffer is configured with a state of completion of processing; and~~ the buffer cache is configured to place the first data buffer on the free data buffer list even though there is an incomplete state of completion of the processing of the first descriptor buffer.

Claim 8. (currently amended) A buffer cache as recited in claim 7, wherein the buffer cache further comprises:

a free ~~descriptor data~~ buffer list is configured with a plurality of positions to identify only descriptor buffers in a free condition; ~~and~~

the configuring of the buffer cache allows the taking of the first ~~second~~ descriptor buffer from any of the positions on the free descriptor buffer list.

Claim 9. (currently amended) A buffer cache as recited in claim 2 ~~1~~, wherein:

the buffer cache is configured with a plurality of data buffers, the plurality including the first data buffer;

the free data buffer list is a linked list and is configured to list a plurality of data buffers each of which is in a free condition; and

the first descriptor buffer is associated with the first data buffer and has a descriptor having values identifying free data buffers that are previous and next relative to the first data buffer associated with the first descriptor buffer.

Claim 10. (original) A buffer cache having an efficient data buffer utilization characteristic, the buffer cache comprising:

a plurality of data buffers; and

a plurality of descriptor buffers, each of the descriptor buffers having a descriptor configured with a plurality of values, the values comprising values representing a data buffer that is free, the values further comprising values representing a descriptor buffer that is free, the free data buffer values being changeable independently of the free descriptor buffer values to identify one of the data buffers that is free at the same time as a descriptor buffer associated with that one data buffer is not free.

Claim 11. (original) A buffer cache as recited in claim 10, further comprising:

a first list header having a first value indicative of one of the free data buffers that is least recently used in the buffer cache, the first list header having a second value indicative of one of the free data buffers that is most recently used in the buffer cache.

Claim 12. (original) A buffer cache as recited in claim 11, further comprising:

a second list header having a third value indicative of one of the free descriptor buffers that is least recently used in the buffer cache, the second list header having a fourth value indicative of one of the free descriptor buffers that is most recently used in the buffer cache.

Claim 13. (currently amended) A buffer cache having an efficient utilization characteristic in response to a plurality of data access requests, comprising:

a plurality of data buffers;

a plurality of descriptor buffers;

one of the descriptor buffers being configured to be associated with one of the data buffers for response to one of the data access requests; ~~and~~

each of the data buffers and descriptor buffers being configured to be separately processed in response to that one data access request;

a free data buffer list configured as a free data buffer link list linking those data buffers that are free; and

the buffer cache being configured to link the one data buffer on the free data buffer list according to a state of completion of processing of only that one data buffer, that linking being independent of a state of completion of processing of the associated descriptor buffer.

Claim 14. (currently amended) A buffer cache as recited in claim 13, wherein:

the buffer cache is further configured with a free descriptor buffer list configured as a free descriptor buffer link list linking those descriptor buffers that are free;

the buffer cache is further configured to determine whether each of the associated data buffer and descriptor buffer is on the respective free data buffer link list or respective free descriptor buffer link list; and

the buffer cache is configured to unlink the free associated data buffer from the respective free data buffer link list and to unlink the free associated descriptor buffer from the respective free descriptor buffer link list for use in the buffer cache to respond to the one of data access requests.

Claim 15. (currently amended) A buffer cache as recited in claim 13, wherein:

the one associated data buffer is a first associated data buffer and the one associated descriptor buffer is a first associated descriptor buffer;

in response to a first data access request, the state of completion of processing of that first associated data buffer becomes complete before the state of completion of processing of that associated descriptor buffer becomes complete;

the buffer cache is further configured with a free descriptor buffer list configured as a free descriptor buffer link list linking those descriptor buffers that are free;

the buffer cache is further configured to determine whether a second descriptor buffer is on the respective free descriptor buffer link list; and

the buffer cache is configured to unlink from the free descriptor buffer link list the second free descriptor buffer for use with the first data buffer to respond to a second data access request for data that is not already stored in the buffer cache.

Claim 16. (currently amended) A buffer cache as recited in claim 15, wherein:

the buffer cache is further configured to terminate the association of the first associated data buffer and the first associated descriptor buffer and to associate the first data buffer and the second descriptor buffer for response to the ~~a~~-second of the data access requests.

Claim 17. (original) A buffer cache having an efficient utilization characteristic, comprising:

a plurality of data buffers;

a plurality of descriptor buffers;

the buffer cache being configured so that any one of the data buffers and any one of the descriptor buffers may be in exclusive association with each other and not with any other respective data buffer or descriptor buffer at the same time;

a search cache algorithm for finding, if there is one, a descriptor buffer corresponding to a specified portion of data and, if there is such corresponding descriptor buffer, also finding a data buffer that is both associated with such descriptor buffer and contains the specified portion of data;

a free data buffer list configured to list the data buffers that are in a free state;

and

a free descriptor buffer list configured to list the descriptor buffers that are in a free state;

the buffer cache being configured to determine whether each of the corresponding descriptor buffer and associated data buffer is on the respective free list.

Claim 18. (original) A buffer cache as recited in claim 17, wherein the buffer cache is further configured to respond to a data access request and in response initiate operations of the buffer cache, the operations being to take the associated data buffer off the free data buffer list and to take the corresponding descriptor buffer off the free descriptor list.

Claim 19. (original) A buffer cache as recited in claim 17, in which the free data buffer list is configured such that substantially immediately upon completion of processing of the data buffer the data buffer is placed on the free data buffer list independently of completion of processing of the descriptor buffer.

Claim 20. (original) A buffer cache as recited in claim 17, wherein the buffer cache is further configured such that the association of the data buffer and the corresponding descriptor buffer is retained even though the free data buffer list lists the data buffer as being free.

Claim 21. (currently amended) A buffer cache as recited in claim 17, wherein:

the buffer cache is configured such that upon the search cache algorithm not finding the descriptor buffer corresponding to the specified ~~specific~~ portion of the data, the buffer cache identifies ~~identifying~~ a free descriptor buffer at the head of the free descriptor buffer list and ~~clearing~~ clears values of that head free descriptor buffer and populates ~~populating~~ the values with respect to the ~~new~~ specified portion of the data; and

the buffer cache is configured to use that populated descriptor buffer with any data buffer on the free data buffer list if no free data buffer is associated with the populated descriptor buffer.

Claim 22. (original) A buffer cache as recited in claim 17, wherein the buffer cache is configured to list the descriptor buffer on the free descriptor buffer list when processing of the descriptor buffer has been completed.



Claim 23. (original) A fault recovery system having a plurality of input-output managers, each of the input-output managers being configured to manage the storage of a portion of a data object to physical memory, for each input-output manager the system comprising:

- a buffer cache comprising a data buffer and a descriptor buffer associated with each other;

- a plurality of the associated data buffers and descriptor buffers;

- a plurality of unassociated descriptor buffers not associated with any of the data buffers;

- the buffer cache being configured to separately process each associated data buffer and to separately process each associated descriptor buffer in the use of the buffer cache;

- the fault recovery system further comprising:

- a free data buffer list configured to identify data buffers that are in a free state;

and

- a free descriptor buffer list separate from the free data buffer list and configured to identify descriptor buffers that are in a free state;

- each buffer cache being further configured to place any respective data buffer on the separate free data buffer list according to the state of completion of processing of that data buffer, the buffer cache being further configured to place any respective descriptor buffer on the free descriptor buffer list according to the state of completion of processing of that descriptor buffer;

- the buffer cache being further configured to break the association of any of the associated data buffers and descriptor buffers to form one or more unassociated descriptor buffers and one or more unassociated data buffers, and to associate one of the unassociated data buffers with one of the unassociated descriptor buffers;

- the respective input-output manager facilitating the storage of any portion of the data object to the physical memory by use of an associated data buffer and descriptor buffer from the respective buffer cache of the respective input-output

manager; and

the buffer cache of the respective input-output manager being configured to cause one of the respective data buffers to be taken from the respective free data buffer list for use with a respective associated descriptor buffer taken from the respective free descriptor buffer list.

Claim 24. (original) A fault recovery system as recited in claim 23, in which the buffer cache is configured so that with one of the associated data buffers on the free data buffer list, the association of the associated descriptor buffer with that data buffer may be retained.

Claim 25. (canceled)

Claim 26. (currently amended) A method as recited in claim ~~25~~ 29, in which:

the configuring operation is independent of a state of processing of the descriptor buffer.

Claim 27. (currently amended) A method as recited in claim ~~25~~ 29, in which:

the configuring operation frees the data buffer immediately upon the state of completion of processing of the data buffer being a complete state.

Claim 28. (currently amended) A method as recited in claim ~~25~~ 29, in which the configuring operation further comprises associating the data buffer and the descriptor buffer with each other separately from a state of processing of the data buffer and a state of processing of the descriptor buffer.

Claim 29. (currently amended) A method of providing efficient buffer cache utilization, comprising the operations of: ~~as recited in claim 25, in which the configuring operation further comprises~~

establishing in the buffer cache a data buffer and a descriptor buffer, the data buffer and the descriptor buffer being configured to be separately processed in the use of the buffer cache;

configuring the buffer cache to free the data buffer according to a state of completion of processing of only the data buffer, wherein that state of completion is independent of a state of completion of processing of the descriptor buffer, the configuring operation further comprising associating the data buffer and the descriptor buffer with each other separately from the freeing of the data buffer.

Claim 30. (original) A method as recited in claim 29, wherein the configuring operation further comprises configuring the buffer cache to retain the association of the data buffer and the descriptor buffer while the data buffer is free.

Claim 31. (currently amended) A method of providing efficient buffer cache utilization, comprising ~~as recited in claim 25, in which the configuring operation further comprises~~ the operations of:

establishing in the buffer cache a data buffer and a descriptor buffer, the data buffer and the descriptor buffer being configured to be separately processed in the use of the buffer cache; and

configuring the buffer cache to free the data buffer according to a state of completion of processing of only the data buffer, wherein that state of completion is independent of a state of completion of processing of the descriptor buffer, the configuring operation further comprising the operations of:

configuring the buffer cache to free the descriptor buffer according to a state of completion of processing of only the descriptor buffer; and

associating the data buffer and the descriptor buffer with each other separately

from each of the ~~configuring~~ configuration operations.

Claim 32. (original) A method as recited in claim 31, in which the operation of configuring the buffer cache to free the descriptor buffer further comprises retaining the association of the data buffer and the descriptor buffer while the descriptor is buffer free.

Claim 33. (currently amended) A method of providing efficient buffer cache utilization, as recited in claim 25, further comprising the operations of:

establishing in the buffer cache a data buffer and a descriptor buffer, the data buffer and the descriptor buffer being configured to be separately processed in the use of the buffer cache;

configuring the buffer cache to free the data buffer according to a state of completion of processing of only the data buffer, wherein that state of completion is independent of a state of completion of processing of the descriptor buffer;

configuring the buffer cache with a plurality of data buffers;

configuring the buffer cache to receive a data access request, wherein a response to the data access request may be made using one of the data buffers associated with a second descriptor buffer with which the one data buffer is not currently associated, the one data buffer being currently associated with a first descriptor buffer; and

configuring the buffer cache to end the current association of the one data buffer and the first descriptor buffer with each other and to associate the one data buffer and the second descriptor buffer with each other.

Claim 34. (currently amended) A method as recited in claim ~~25~~ 29, wherein the method further comprises the operation of configuring the buffer cache with a plurality of the data buffers and a plurality of the descriptor buffers; and wherein:

the operation of configuring to free the data buffer comprises configuring a

free data buffer link list linking those data buffers that are free, the linking of the free data buffers is according to the state of completion of processing of only each respective data buffer;

the buffer cache is further configured to free the descriptor buffers by configuring a free descriptor buffer link list linking those descriptor buffers that are free, the linking of the free descriptor buffers is according to a state of completion of processing of only each respective descriptor buffer;

the buffer cache is further configured to unlink from the data buffer link list those data buffers that are to be used to cache data and to unlink from the descriptor buffer link list those descriptor buffers that are to be used to cache data; and

the buffer cache is further configured with a link list header corresponding to each of the free data buffer link list and the free descriptor buffer link list, wherein the respective link list header is configured to identify the free data buffer that is at a head of the free data buffer link list and the respective descriptor buffer link list header is configured to identify the free descriptor buffer that is at a head of the free descriptor buffer link list.

Claim 35. (original) A method as recited in claim 34, wherein the buffer cache is further configured to respond to a data access request using the data buffer link list header to identify the head data buffer to unlink from the head of the data buffer link list and using the descriptor buffer link list header to identify the head descriptor buffer to unlink from the head of the descriptor buffer link list.

Claim 36. (original) A method of efficiently managing the storage of data in a buffer cache in response to successive data access requests, comprising the operations of:

receiving a first data access request that requires access to data in the buffer cache;

searching the buffer cache for a descriptor buffer corresponding to the data to be accessed in the buffer cache;

if the searching does not find that corresponding descriptor buffer, determining whether there is any free data buffer in a free data buffer list in the buffer cache, wherein the free data buffer list is separate from a free descriptor list;

if there is any free data buffer in the free data buffer list in the buffer cache, searching the buffer cache for a free descriptor buffer in a free descriptor buffer list that is separate from the free data buffer list; and

if there is a free descriptor buffer in the free descriptor buffer list, taking each of the free data buffer and the free descriptor buffer from the respective free buffer list for use in response to the first data access request.

Claim 37. (original) A method as recited in claim 36, comprising the further operations of:

updating a value of a descriptor of the taken descriptor buffer to point to the taken data buffer to associate the taken data buffer with the taken descriptor buffer; and

responding to the first data access request by writing data into the taken data buffer or reading data from the taken data buffer.

Claim 38. (original) A method as recited in claim 36, wherein a state of operation of the taken data buffer for data caching is indicated by descriptor values of the associated taken descriptor buffer, the values comprise a not-dirty descriptor value and a zero count descriptor value, the method comprising the further operation of:

preparing the buffer cache for further caching operations using that taken data buffer, the preparing comprising the operation of freeing that taken data buffer by returning that data buffer to the free data buffer list independently of a state of operation of the taken descriptor buffer for data caching.

Claim 39. (new) A buffer cache as recited in claim 6, the buffer cache further comprising:

the buffer cache being further configured with the second descriptor buffer having a descriptor having a fifth value that identifies the second descriptor buffer as a particular descriptor buffer that is on the free descriptor buffer list, the fifth value being independent of the third value; and

the buffer cache being further configured to change the first value of the descriptor of the first descriptor buffer to make an association between the second data buffer and the first descriptor buffer.